السلام عليكم ....

في هذه المحاظرة سنتعرف على موضوع جديد وهو ما يسمى بـ A/D Converter

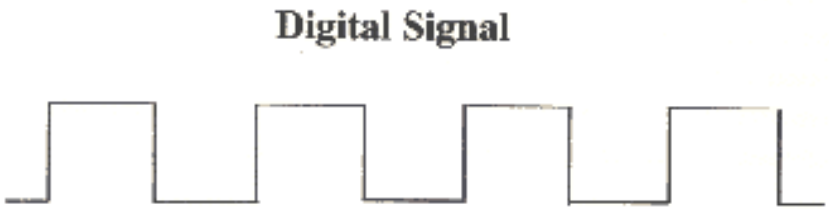
فما هو A/D Converter ؟ ولماذا نحتاجه ؟

فمن المعروف ان المايكروكنترولر يتعامل مع اشارات الـ Digital ولا يتعامل مع اشارات الـ Analog فكيف سأتمكن من قراءة هذا النوع من الاشارات ؟ ... يتم ذلك من خلال استخدام A/D Converter

وهو اختصار لـ Analog to Digital Converter

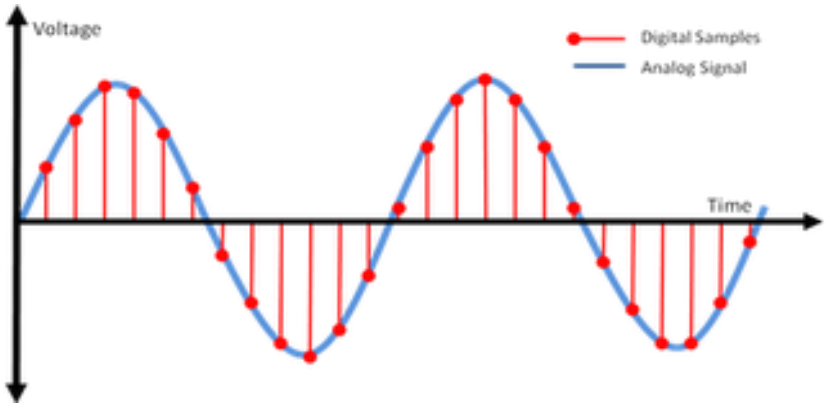
في بداية الموضوع لنطرح هذا السؤال ... ما هو الفرق بين اشارات الـ Analog واشارات الـ Digital ؟؟

وللاجابة لننظر على الصورة التالية



فإشارة الـ Digital : تبقى ثابته لمده زمنيه معينه ... وفي المايكروكنترولر تأخذ قيمتين فقط وهما ٥ فولت و صفر فولت

اما في اشارات الـ analog : فلها قيمه مختلفه عند كل لحظة زمنيه ... وهذا هو الفرق الجوهري بين الاشارتين



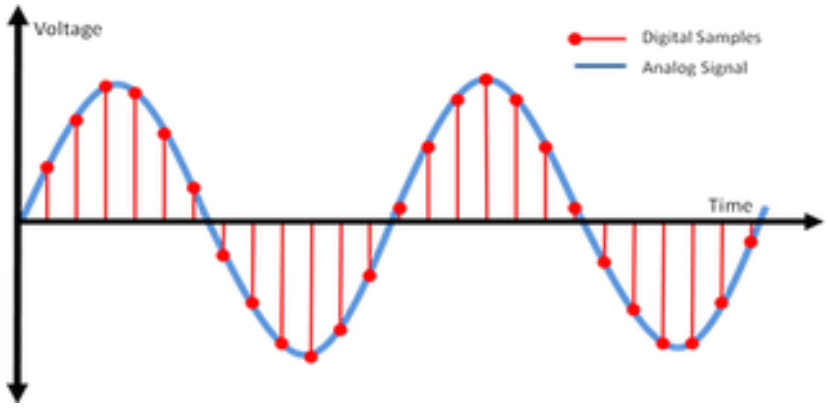
والان سندخل على مبدأ عمل الـ ADC .... وماذا يعني عندما نقول ان هذا الـ ADC هو 10 Bit ؟؟

يتلخص عمل ADC بالتالي:

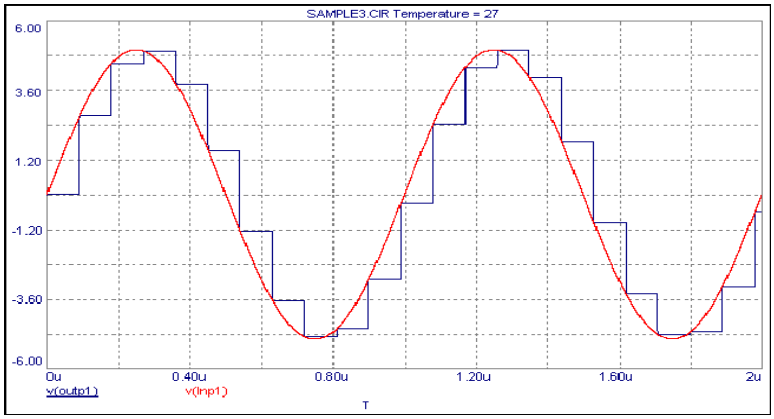
١ -يقوم ADC في البداية بأخذ عينات من الإشارة الأصلية ويثبتها خلال فترات زمنية ثابتة ..
٢ -يتم تقسيم المسافة على كامل مجال الإشارة المطلوبة إلى مستويات ثابتة..
٣ -يتم تحديد المستوى الذي تنتمي إليه الإشارة عند العينة المطلوبة.
٤ -يتم إخراج قيمة المستوى المطلوب على المخارج PORT.

نلاحظ فان عمل ADC يتلخص في مجموعة خطوات بسيطة (لا تقلق إذا لم تفهم ما تعنيه كل خطوة لأننا سنتحدث عن كل خطوة بالتفصيل).. الآن سنبدأ شرح كل خطوة بالتفصيل :

١ـ أخذ العينات من الاشارة الاصلية وتثبيتها : في بداية العملية يتم اخذ عينات تعبر في شكلها عن الاشارة الاصلية ... كما في الشكل التالي: (العينات هي باللون الاحمر)



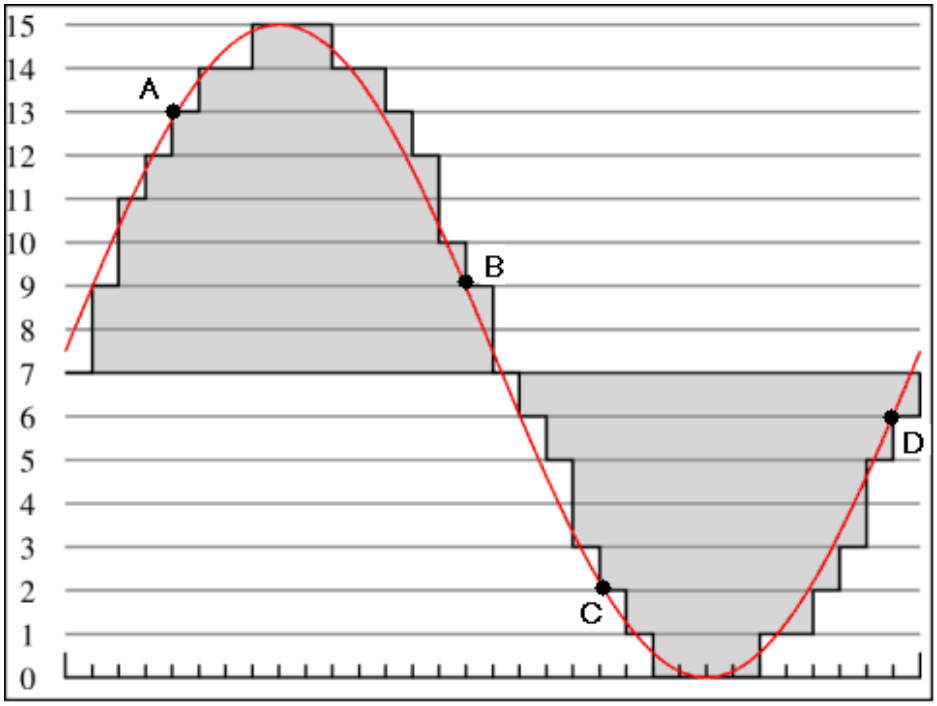
الان يتم اخذ العينة وتثبيت القيمة لهذه العينة خلال فترة عملية التبديل كما في الشكل التالي:



ملحوظة:

كلما زاد عدد العينات المأخوذة نلاحظ اقتراب شكلها اكثر من الاشارة الاصلية وهذا يسمى تردد التقطيع للاشارة..طبعا اذا قل عدد العينات عن عدد معين فان الاشارة سيحدث لها تشويه وستختلف عن الاشارة الاصلية لذلك يجب ان يكون عدد العينات كافي ليشابه الاشارة الاصلية.

٢ـ تقسيم المسافة على كامل مجال الاشارة الى مستويات: الان بعد ان تم تثبيت قيمة العينة خلال الفتره فانه يتم تقسيم المجال العمودي الى مستويات ثابتة كما في الشكل التالي:



فهنا تم تقسيم المدى العمودي الى 16 مستوى من(0 – 15) حيث يتم تحديد العينة ورقم المستوى المقابل لها ..فكما نرى مثلا ان العينة (A) لها قيمة مستوى مساوية لـ 13 ، والعينة (B) لها قيمة مساوية لـ 9 ، والعينة (C) لها قيمة مساوية لـ 2 ، والعينة (D) لها قيمة مساوية لـ 6.

الان يتم اخراج قيمة المستوى المكافىء على اطراف خرج المبدل من تماثلي الى ديجيتال.

الان لو سألتك سؤالا : ما نوع ADC السابق ، هل هو ١٦ بت كما رأيت من الشكل السابق ؟؟؟

طبعا لا فليست هذه طريقة الحساب بالنسبة ADC ... بما اننا نتعامل مع النظام الرقمي(الديجيتال) هنا في خرج الـ ADC فان عملية التعبير يجب ان تكون تابعه لنظام الديجيتال .وبالتالي نقوم بمعرفة عدد مستويات الخرج وهي هنا في حالتنا 16 مستوى وبالتالي لنعرف نوع المبدل نستخدم العلاقة التالية:

$$2^n = 16$$

حيث يعبر الرقم 2 عن النظام الرقمي ... اما n عدد البتات ... اما الرقم 16 فهو عدد المستويات وفي حال كان عدد المستويات غير ذلك نقوم باستبدال الرقم 16 في العلاقة السابقة بالعدد الجديد ونحسب اعتمادا عليه

الان نستطيع بسهولة حساب حسب عدد البتات من العلاقة السابقة وهي تساوي n=4 بت ، يمكن حساب عدد بتات خرج أي ADC من العلاقة السابقة بمعرفة عدد مستوياته ، او بالعكس يمكن معرفة عدد مستويات المبدل من خلال عدد البتات للخرج (علاقة تبادلية سهلة)

ملاحظة : بزيادة عدد مستويات الخرج للمبدل (أي زيادة عدد بتات الخرج للمبدل) فان المبدل يصبح أكثر دقة لأن الاشارة تقترب اكثر من الشكل الاصلي .

الان ننتقل الى السجلات (Registers) التي تتحكم بعملية التبديل من انالوج الى ديجيتال وهي :

ADCON0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

1

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits
000 = channel 0, (AN0)
001 = channel 1, (AN1)
010 = channel 2, (AN2)
011 = channel 3, (AN3)
100 = channel 4, (AN4)
101 = channel 5, (AN5)
110 = channel 6, (AN6)
111 = channel 7, (AN7)

2

Note: The PIC18F2X2 devices do not implement the full 8 A/D channels; the unimplemented selections are reserved. Do not select any unimplemented channel.

bit 2 **GO/DONE:** A/D Conversion Status bit
When ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
0 = A/D conversion not in progress

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit
1 = A/D converter module is powered up
0 = A/D converter module is shut-off and consumes no operating current

Figure 2.42: ADCON0 register

يتكون من مجموعة بتات من اجل السيطره على عملية التحكم في المبدل :

- البت رقم ٧ والبت رقم ٦ : يتحكمان في تردد المبدل حيث يمكن اختيار عدة ترددات حسب الموجود في الجدول رقم ١ الملون باللون الاحمر . (طبعا هذا يتم بمساعدة البت رقم ٦ في المسجل ADCON1) .

مهم :قد يطرح احدكم سؤال ويقول هل يمكن اختيار أي تردد من الجدول السابق ام هناك طريقة معينة ؟؟

الجواب سيكون طبعا لا .. فهناك شي يجب ان ننتبه له عند عملية التحويل ان ADC الداخلي يعمل بسرعة معينة وهذا موجود في الداتا شيت كما في الصورة

TABLE 17-15: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D Clock Period	PIC16F87XA	1.6	—	—	µs	Tosc based, VREF ≥ 3.0V
			PIC16LF87XA	3.0	—	—	µs	Tosc based, VREF ≥ 2.0V
			PIC16F87XA	2.0	4.0	6.0	µs	A/D RC mode
			PIC16LF87XA	3.0	6.0	9.0	µs	A/D RC mode

ان ADC الداخلي للانواع التي تحمل الرقم PIC16F87XA مثل PIC16F877A تحتاج الى زمن قدره ١,٦ ميكروثانية ليقوم بتحويل بت واحد وبالتالي يجب ان نختار قيمة التردد بحيث يكون الزمن كافيا (طبعا التردد هو معكوس الزمن للاشارة F=1/T، وبالتالي من خلال التردد نحصل على الزمن T للتردد المقصود) .

فاذا كان الزمن الناتج من التردد على الاقل يساوي ١,٦ ميكرو ثانية فاختيارك صحيح (لا يتم اختيار زمن اقل لان الزمن سينقضي قبل ان تنتهي عملية التبديل للبت الواحد ،ايضا لا يتم اختيار زمن اكبر من ١٠ ميكرو ثانية لانه سيؤدي الى استهلاك كبير للطاقة).

- البت رقم ٣ والبت رقم ٤ والبت رقم ٥ : تستخدم من اجل تحديد أي من الاطراف الثمانية سوف يستخدم كمدخل انالوج بحيث يعطي اشارةته الى المبدل الداخلي.

- البت رقم ٢ : عندما نريد ان نبدا عملية التحويل يجب ان نفعل هذا البت (نضع فيه القيمة ١) بحيث يبدأ زمن الاكتساب (الزمن اللازم حتى يشحن المكثف على مدخل المبدل).. في حال انتهاء عملية التبديل فان هذا البت يتحول تلقائيا الى الصفر ..وبالتالي يمكن ان نتفحص هذا البت لنعرف اذا كان المبدل قد اتم عمله ام لا حتى نستطيع الانتقال الى تحويل اشارة انالوج اخرى .

- البت رقم ١ : غير مستخدم وعادة نضع القيمة صفر فيه .

- البت رقم ٠ : يستخدم من اجل تفعيل خاصية التحويل من انالوج الى ديجيتال (تشغيل المبدل) ، أي يجب ان نفعل هذا البت حتى يبدا المبدل بالعمل ، ثم بعدها نفعل البت رقم ٢ من اجل ان يبدا المكثف بالشحن وتبدأ عملية التحويل .

كما لاحظنا فان هذا السجل هو المسؤول عن عملية التبيدل كلها ، و يجب ان يتم التحكم فيه بمعرفة ودراية والا فان هناك احتمال خطا او اساءة استعمال لخواص هذا المبدل في أي عملية نريد تنفيذها .

ADCON1

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.
0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

bit 6 **ADCS2:** A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-4 **Unimplemented:** Read as '0'

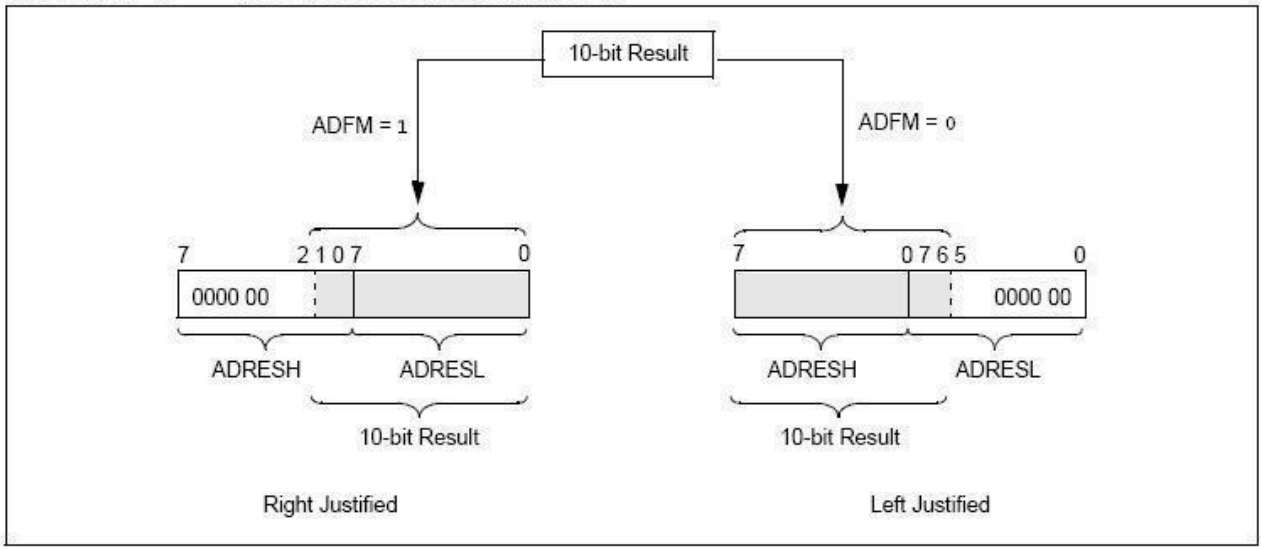
bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = # of analog input channels/# of A/D voltage references

- البت رقم ٧ : ان نتيجة التبديل هي عبارة عن ١٠ بت ، وتخرج نتيجة المبدل الى مسجلين هما (ADRESH) والمسجل (ADRESL) وبما انهما مسجلان فينتج لدينا ١٦ بت ، اذن يبقى هناك ٦ بت فارغة وغير مستخدمة وهذا البت يتحكم فيما اذا كانت هذه البتات ال ٦ الفارغة ستكون في المسجل (ADCON0) او المسجل (ADCON1) .. كما في الشكل التالي:

FIGURE 11-4: A/D RESULT JUSTIFICATION



- البت رقم ٦ : كما قلت سابقة يستخدم مع البت رقم ٧ والبت رقم ٦ في المسجل السابق ADCON0 لاختيار تردد عمل المبدل ..

- البت رقم ٥ والبت رقم ٤ : غير مستخدم وعادة نضع القيمة صفر فيه .

- البت رقم ٣ والبت رقم ٢ والبت رقم ١ والبت رقم ٠ : تستخدم هذه البتات من اجل التحكم باطراف الانالوج ، فكما نعلم يمكن ان نستخدم الاطراف كمداخل ديجيتال او مداخل انالوج ، هذه البتات تحدد نوع الاطراف المستخدمة حسب الجدول الازرق ، طبعا يتم اختيار اقرب ترتيب لما نريد نحن لاننا قد لا نجد بالضبط جميع الاحتمالات لانواع هذه الاطراف .

ملاحظة مهمة : في الجدول نلاحظ وجود (VREF+) و (VREF-) وهذا الاطراف يمكن ان نربط بها مجال جهد الدخل بحيث يتغير عن (٠ - ٥ فولت) فيمكن مثلا في حال كان لدينا سنسور معين يخرج جهد بقيمة من (٢-٤ فولت) فيمكن ان نحصل على قيمة حساسية اكبر للـ ADC لان المجال اصبح ٢ فولت بدلا من ٥ فولت وبالتالي دقة اكبر وتحكم اكبر في مشروعنا المطلوب .

لاحظ الجدول في الاعلى .... وأوجد الرقم الذي يجب ان يحتوي عليه الرجستر ADCON1 كي نستخدم جميع مداخل الانالوج كمداخل ديجيتال .... مذا تجد؟؟

ستلاحظ ان النتيجه هي : ADCON1 = 0x06 أو ان يكون ADCON1 = 0x07

وهذه الجمله استخدمناها في البرمجه في احدا الدروس السابقه وذكرنا بأننا سنفصلها لاحقاً والشرح السابق كان تفصيلاً لها .

والان سنبدا اولى الخطوات العملية في كتابة البرامج من اجل التعامل مع هذه الخاصية وتوضيح الاوامر المستخدمة لذلك في لغة الميكروسي

ما يميز الميكروسي ان عملية قراءة الاشارات الانالوج تتم باستخدام امر واحد فقط يقوم بمعظم الاعدادات والترتيبات لوحده بحيث ما عليك الا ان تقرا القيمة وهو سيقوم بالباقي. . هذا الامر هو التالي:

```
ADC_Read (X)
```

حيث(X) هو رقم الطرف الذي نستخدمه كمدخل انالوج في هذه اللحظة.

يتم حفظ القيمة الرقمية المكافئة في متغير مكون من ١٦ بت (int unsigned) حتى نستطيع ان نتعامل مع المبدل ذو الـ 10 bit

الاعدادات العملية في أي عملية تحويل من انالوج الى ديجيتال:

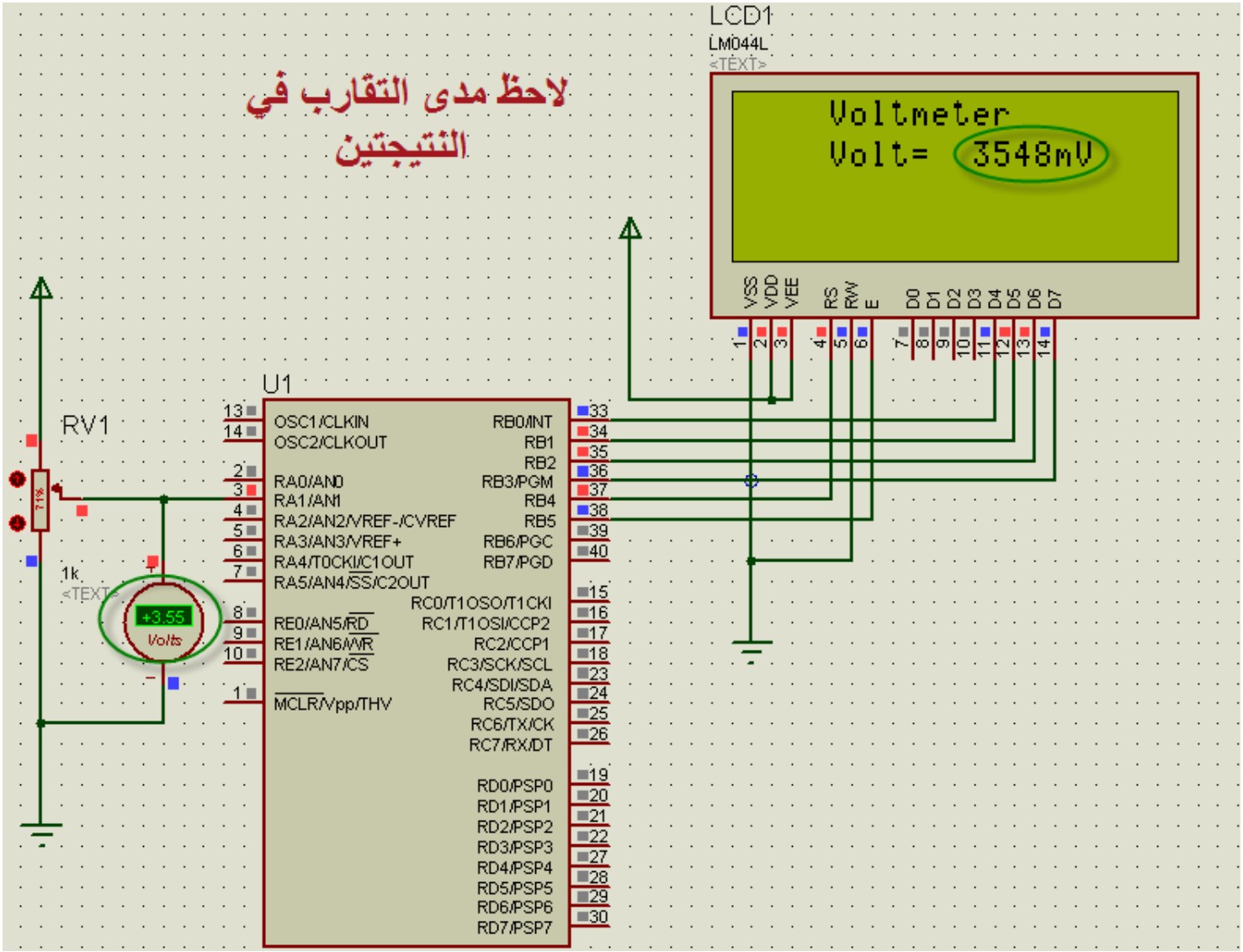
١ ـ تعريف الاطراف التي نريد ان نقرا منها اشارات الانالوج كمداخل من خلال المسجل TRIS
٢ ـ تحديد نوع الاطراف (انالوج / ديجيتال) ، بالاضافة الى تحديد مكان البتات الفارغة من خلال المسجل ADCON1
٣ ـ قراءة الاشارة الانالوج في اللحظة التي نريدها وحفظها في متغير من نوع (unsigned int).

ملاحظة مهمة جدا : ان الامر ( ) ADC_READ يقوم بتغيير الاعدادات في مسجل التحكم ADCON0 لذلك يكفي ان نضع القيمة المطلوبة في المسجل ADCON1 ونترك كامل عملية التحكم للامر في الميكروسي ليقوم بعمليات الضبط ..للعلم يقوم الميكروسي بوضع زمن ثابت للمبدل وهو المعتمد على الهزاز الداخلي للبك RC الداخلي وهو ياخذ زمن تقريبا ٢-٦ ميكروثانية في عملية اتحويل .

والان ... هل خطر على بالك ان تصمم مقياس فولتميتر بسيط تقيس به الجهود المستمرة ؟؟

اذا نحن على وشك تصميم ابسط فولتميتر يقيس الجهود من ٠ - ٥ فولت .. ويعرضها على شاشة LCD وميزة ان هذا الفولتميتر هو من تصميمنا نحن.

وسنستخدم الدائره الكهربائية التالية التي التقطها اثناء عمل الدائره



والان للننظر الى الكود في الصفحه التالية .... وقرأه سطر سطر

```
// LCD module connections
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// End LCD module connections
//int V;
float V;
char Volt[6];
void main ()
{
 lcd_init();
 lcd_cmd(_lcd_clear);
 Lcd_Cmd(_LCD_CURSOR_OFF);
 adc_init();
 while(1){
  lcd_out(1,5,"Voltmeter");
 lcd_out(2,5,"Volt=");
V=adc_read(1);
V=(V*5000)/1023;
inttostr(V,Volt);
lcd_out_cp(Volt);
lcd_out_cp("mV");
delay_ms(1000);
lcd_cmd(_lcd_clear);
        }

}
```

هذا الجزء خاص في شاشة الـ LCD وتم شرحه في المحاضرة السابقة

تعريف متغيرين لحفظ القيم فيهما احدهما int والاخر من نوع charلان الـ LCD لا تقبل الـ int

تهيئة الـ ADC للعمل

أمر البيك بالقراءة من مدخل الـ Analog1 والموصل كما في الدائرة في الاعلى

أمر مهم جداً لتحويل من int الى char

زمن تأخير مهم جدا لاعطاء الـ LCDفتره لعرض القيم

تابع الكود و اقرأ الملاحظات الموجودة .... ونحن بأنتظار الاستفسارات .

كما وسيتم ارفاق ملف المحاكاة والبرنامج للقيام بعمل تعديلات على الكود وملاحظة النتيجه ... لتتمكن من معرفة اهمية كل جمله من جمل الكود

انتهت المحاظرة السابعة